

2014

University of North Carolina Wilmington
Master of Science in
Computer Science and Information Systems
Proceedings

<https://csbapp.uncw.edu/mscsis>

A CROSS-PLATFORM MOBILE HEALTH APPLICATION

Justin McNeil

A Capstone Project (or Thesis) Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Computer Science
Department of Information Systems and Operations Management

University of North Carolina Wilmington
2014

Approved by

Advisory Committee

Dr. Cara Sidman

Dr. Thomas Janicki

Dr. Wayland Tseh

Dr. Devon Simmonds (Chair)

Accepted By

Dean, Graduate School

Abstract

A Cross-Platform Mobile Health Application. McNeil, Justin, 2014. Capstone Paper, University of North Carolina Wilmington.

The purpose of this paper is to research and explore the processes of developing a cross-platform mobile application. Additionally, the paper explores the advantages and disadvantages of the hybrid approach to mobile development. The paper describes the development process of the mobile application, Walk Tracer, that calculates caloric expenditure using the American College of Sports Medicine metabolic formulas. Walk Tracer gives users the ability to measure the calories used during walk or run exercises. Walk Tracer will provide the user information on their past workouts and use the google maps API to display the user's route. This app was developed using J query mobile, HTML5, local storage, JavaScript and CSS3 web technologies. The paper includes an analysis of competitors, an overview of different mobile development strategies and suggestions for the future implementation of Walk Tracer.

Table of Contents

	Page
Chapter 1: Introduction.....	1
1.1 Overview.....	1
1.2 Project Goals.....	3
Chapter 2: Review of Literature Review and Analysis	4
2.1 Mobile App Reviews	4
2.1.1 Runtastic	4
2.1.2 Calorie Counter.....	4
2.1.3 MapMyRun	5
2.1.4 RunKeeper	6
2.1.5 App Comparisions	6
2.2 Definitions of Terms.....	6
2.3 Developing a Mobile Application.....	7
2.3.1 Native Mobile Applications.....	7
2.3.2 Mobile Web Application.....	9
2.3.3 Hybrid Application	10
2.4 Going Hybrid	11
2.4.1 PhoneGap.....	11
2.5 Naming the Application.....	12
Chapter 3: Methodology	14
3.1 Overview.....	14

	2
3.2 Use Case Diagram.....	15
3.3 User Interface Prototype	15
3.3.1 Home Screen Page	16
3.3.2 Enter A Walk or Run	17
3.3.3 Profile Page.....	19
3.3.4 Food Log Page	20
3.3.5 Progress Page.....	21
3.4 Changes to the Prototype	23
3.5 Technology Used	23
3.5.1 JQuery Mobile	23
3.5.2 JavaScript.....	24
3.6 Software Evaluation and Testing.....	24
3.6.1 Testing Procedures.....	25
Chapter 4: Outline of Completed Project.....	26
4.1 Setting up the Environment	26
4.2 Getting Started-Homepage.....	26
4.3 Profile page	26
4.4 Manual Run and Handling Local Storage.....	27
4.5 The Road to Data Validation	28
4.6 PhoneGap Geolocation	30
4.6.1 Adding Google Maps.....	31
4.7 History and Progress Page	32

Chapter 5: Conclusions and Future Work.....	33
5.1 Project Deliverables.....	33
5.2 What I Learned	34
5.3 Future Work	34
References.....	35
Appendixes	
A. App Use Cases	37
B. Project Schedule.....	47
C. American College Association of Sports Medicine Formulas.....	49
D. Project Prototype.....	51
E. Project Screen Shots	55

Chapter 1: Introduction

1.1 Overview

This project, which was developed as the WalkTracer Application, was selected due to an interest in mobile application (app) development. It had its beginning with the help of members of various departments at the University who assisted in locating a client that expressed an interest in having a mobile app developed. The client wanted a health mobile application designed that calculated the calories expended, while the user walked or ran, based on the American College of Sports Medicine (ACSM) caloric expenditure equations [1].




The mobile application market is flooded with an abundance of mobile health apps. However, according to the “*Consumer Health Apps for Apple's iPhone Report*” published in 2011, there were over 13,000 mobile health apps on iOS alone and there remained an enormous market for mobile health apps [18]. According to the Pew Research Center’s Internet and American Life Project’s mobile health survey released in November 2012, nineteen percent of mobile phone users had downloaded a health app [9]. Furthermore, according to another study also by Pew Research Center’s Internet and American Life Project, 56 percent of all American adults own a smart phone¹ [10]. A large number of smart phone users also use mobile health applications; therefore there is a market for another mobile health application.

Walk Tracer uses unique, reliable and proven ACSM equations in its caloric expenditure calculations. Many apps on the market do not reveal the equations they used to calculate metabolic caloric expenditures. Additionally, many current apps gave inconsistent results by calculating different caloric expenditures for the exact same

¹ Smartphone is any phone identified as an iPhone, Android phone, Blackberry or Windows phone

workout. These discrepancies were demonstrated when running workout results were manually inputted into three different mobile health apps: Runtastic, RunKeeper and MapMyRun. From the results obtained, the calories burned during the workout were able to be determined. Table 1 illustrates the different results of these entries.

Table 1 *Calories Burned Comparison*

			
Miles	5	5	5
Duration	1 hour	1 hour	1 hour
Pace	12:00	12:00	12:00
Weight of Runner	180 pounds	180 pounds	180 pounds
Calories Burned	729	653	711

While the discrepancies in Table 1 may have seemed insignificant, they were not since weight loss occurs when the body expends more calories than it consumes [27]. Thus the accurate calculation of calories burned during a walk or run is extremely important. By calculating the number of calories consumed, in addition to measuring the number of calories burned during a walk or run, one can accurately determine weight loss progression. Also, by using accurate caloric expenditure equations, one can determine the length of time it will take to reach a desired weight. Therefore, if individuals were to inaccurately calculate the number of calories expended during a walk or run, they may not reach their desired weight loss goals. In order to give an accurate calculation of the

calories expended during walking and run exercises, Walk Tracer was developed using the metabolic equations formulated by the American College of Sports Medicine [1].

The American College of Sports Medicine was founded in 1954 and is currently the largest sports medicine and exercise science association in the world. The organization has over 45,000 members and represents over 70 occupations within the exercise science and sports medicine field. The goal of the ACSM is to “promote fitness, health and quality of life through research, education and practical applications of sports medicine”[8]. ACSM certifications are the gold standard certifications in the exercise science and sports medicine field [8].

1.2 Project Goals

There were many questions to consider in the development of Walk Tracer. One of the goals in this project was to answer the following questions: How does one begin making a mobile application? On what mobile platform would this application be developed? What makes a good mobile health application? Was there an efficient way to put this application on multiple platforms, without having to develop the application for each platform? Furthermore, with this being the first version of the mobile application, what would make future updates and upgrades less difficult?

The deliverable of this project include a working Android application prototype based on the client’s requirements. Additionally, the client was provided with well-documented code in order to assist any future developers who work on this application. The overall purpose of this project was to learn an efficient way to develop a mobile application across multiple mobile platforms and to satisfy the client by providing a competitive mobile health application.

Chapter 2: Review of Literature Review and Analysis

2.1 Mobile App Reviews

A review of current health and exercise mobile applications was completed in order to assess what should go into a health and exercise mobile application in order to make it successful.

2.1.1 Runtastic

Runtastic developed by MyFitnessPal is an app available for Android and iPhone. Runtastic uses GPS to map a user's various exercise activities including running, walking and jogging. The app tracks a user's time, distance, and speed during the cardio workout process. The goal of the app is to assist users in reaching their exercise goals through tracking all the details of their workouts and showing their progression. Users of the app are also able to share the results of their workouts with friends [14].

Runtastic features include a personal training diary and metrics. The app also provides users with graphs that track their pace, speed and heart rate information from a walk, run, or jog. While the app does use GPS to track the distance of a run, jog, or walk, it also allows for manual entry of all workout data. The app provides monthly metrics for distance, number of workouts, and the calories burned during those workouts [14].

2.1.2 App Reviews Calorie Counter

The calorie counter and food diary by MyNetDiary is a calorie tracking app that assists users in diet planning. Calorie Counter is available on the Android and iPhone platforms. The app includes a database with over 530,000 different foods and their nutritional value. Users are able to update that database by entering their own foods. Daily water consumption is also tracked using this app.

The calorie counter food diary allows users to enter their target weight. After entering a target weight, users are asked to enter a target date, the deadline for reaching their target weight, or a weight loss rate. The app will create a weight loss plan based off the inputted information. As the user updates his or her progress while following the plan, the app will adjust its plan accordingly [15].

2.1.3 App Review MapMyRun

MapMyRun, developed by MapMyFitness, is an app that allows users to map out the route of their walk or run. As the user works out, the app calculates the route, mileage, minutes per mile, and calories burned. However, the application does not give any information as to how it calculates calories expended during a workout. MapMyRun is available on the iPhone, Android, and Blackberry mobile devices. A user's MapMyRun data is synchronized with a user account on the MapMyRun website [16].

MapMyRun can either be downloaded for free or users can pay \$2.99 for the Pro version. The free version of the app contains ads at the bottom of every screen in addition to a full-page ad that appears each time a user goes to a different screen. A benefit of MapMyRun is that it is able to run in the background of a mobile device which allows users to play music or perform other operations on their mobile device while the MapMyRun application is in use [16].

MapMyRun has a database full of the nutritional values for various foods. This feature of the MapMyRun application allows users to enter and obtain nutritional value of a food. The user is able to keep track of foods consumed in a given period of time. MapMyRun will let users know if they are burning more calories than they take in [16].

2.1.4 App Review RunKeeper

RunKeeper is a mobile app developed by FitnessKeeper Inc. that tracks a user's runs, walks, bike rides, and hikes using the GPS in their phone. This app is available on the Android and iPhone Platforms. RunKeeper has been downloaded by over 20 million people. It allows users to see detailed statistics about their pace, distance, and time. The app allows users to set goals and targets, and then notifies the user when those goals or targets have been achieved. Additionally, RunKeeper notifies users when they hit their new personal best or reach new milestones. The app keeps track of calories burned during various workouts. RunKeeper is able to run in the background of the phone and allows users to use other apps while this app is running. This app allows users the convenience of logging in with Facebook, giving users the capability to share their activities and achievements with their Facebook friends [17].

2.1.6 App Comparisons

While research was conducted on many different mobile health apps, there were a number of similar features that were present in all the mobile health apps. RunKeeper, MapMyRun and Runtastic all gave the user the ability to record the calories consumed in a given day. MapMyRun, Runtastic and RunKeeper allowed the user to utilize the phone's GPS while walking or running to record the distance that the user traveled. Additionally, those apps allowed for manual entry of a walk or run. All of the apps recorded users' progress while using the app, and allowed them to bring up the history of their progress. After RunKeeper, MapMyRun and Runtastic were reviewed, the features that would be included in Walk Tracer were determined.

2.2 Definition of Terms

This section will be used to define terms that are used throughout this document. Words and abbreviations related to the field of exercise science will be defined below

alphabetically.

API. Application Program Interface consists of routines, and tools for building software applications.

CSS. Cascading Style Sheet is a style sheet language used for designing the look and formatting of a document that is written in a markup language. [4]

Exercise. Exercise is a type of physical activity consisting of plan, structured, and repetitive bodily movement done to improve or maintain one or more components of physical fitness [1].

Physical Activity. Physical Activity is defined as any bodily movement produced by the contraction of skeletal muscles that result in a substantial increase over resting energy expenditure [1].

VO_{2max} . VO_{2max} is the maximal volume of oxygen consumed per minute. It is the point where your body cannot increase the amount of oxygen it is consuming [2]. The VO_{2max} is used to calculate caloric expenditure during an exercise.

Platform. Platform can be described as a device is operating system. Some examples of mobile platforms are Windows phone, Android, iPhone OS.

2.3 Developing a mobile application

At the start of the project, the first question was on what platform would the application be developed? Additionally, once the platform was selected, what programming language was required to develop applications on that platform? Mobile applications can be developed as native, web, or hybrid applications. Each method of developing mobile applications has its advantages and disadvantages.

2.3.1 Native Mobile Applications

One way of developing mobile applications is to develop it as a native mobile

application. Native application development implies that an application is developed specifically for a particular platform or device [3]. The applications are coded in the specific programming language of a device. For example, Objective C is the programming language for iOS, as Java is the programming language for Android. Native applications are downloaded directly to a user's device and, depending on the application, may not even require internet access to use the application.

There are many advantages to developing native mobile applications. Native mobile applications are able to take full advantage of device specific features such as the camera, gestures, GPS, the accelerometer, etc.[6]. Additionally, native applications contain fluid animation and fast graphics which make it the ideal method for developing mobile games. Also, setting up the development environment to create a native application is pretty straight-forward. One simply uses the tools required by the platform to develop an application for the device [7].

The disadvantage of native applications is that they are more difficult to develop because they require high level knowledge of a specific programming language for a device. *Table 2* shows the different development requirements for the various mobile platforms. In addition, developing an application across multiple platforms would require the developer to know multiple programming languages or have multiple developers. Furthermore, if one develops an application across multiple platforms, then there are many sets of codes, making updates to an application difficult. The update must be developed, tested, and distributed for each mobile platform.

Table 2 Development Requirements for Different Mobile Platforms [19]

Mobile OS	Operating System	Software/IDEs	Programming Language
iOS	Mac only	Xcode	Objective C
Android	Windows/Mac/Linux	Eclipse/Java/Android Development Tool (ADT)	Java
BlackBerry	Windows mainly	Eclipse/JDE, Java	Java
Symbian	Windows/Mac/Linux	Carbide.c++	C++
WebOS	Windows/Mac/Linux	Eclipse/WebOS plugin	HTML/JavaScript/C++
Windows 7 Phone	Windows mainly	Visual Studio 2010	C#, .NET, Silverlight or WPF

2.3.2 Mobile Web Application

Mobile applications can also be developed as HTML5 web applications.

Technically, web apps are not real applications; they are actually websites that look and feel like native applications [11]. Web applications take the user to a mobile version of a website and are not downloaded to the user's device. These applications are run in browsers and are usually hard to distinguish from a native app. Mobile web applications are able to hide browser buttons or bars, and due to browser caching, some features of mobile web applications are even available for use off-line.

There are a couple of ways to handle building a mobile website. A developer who already has a website can make a separate mobile version of its website that has its own unique URL. Users who visit the website with a mobile device will be automatically redirected to the mobile version of the website. Another way to build a mobile website is to use a responsive website design to create a website that will change based upon the browser's window size. [13]

Undoubtedly, there are many advantages to developing web applications for mobile use. For instance, a single mobile website can be used across many different

mobile devices. The developer does not have to submit the mobile app to the various app market places. Additionally, a mobile website is instantly accessible to developer's users when the website is deployed because users are not required to download and install the application. Furthermore, a mobile web app is easy to update or upgrade because all that is needed is one update. There is no requirement to update the app for each individual device platform.

While there are many advantages to developing a mobile web application, one must also consider the drawbacks. Mobile web apps are not able to access specific device features such as user context, file uploading, push notifications, the camera, etc. Also, a mobile web app is at the mercy of whatever network is used to access it, and network access and speed varies from location to location. If a user were to have unreliable coverage, then the performance of the app would be greatly diminished.

2.3.3 Hybrid Application

Another way mobile applications can be developed is as hybrid applications. As the name implies, hybrid mobile applications combine features from both native and web applications. It allows users to develop applications using web technologies such as HTML5, JavaScript, and CSS, and allows each mobile device to use its own browser to display the application. A native wrapper is compiled around the HTML, CSS, and JavaScript which allows the application to interact with the device's native APIs. This allows for use of the specific device features such as a camera, GPS, touch features, etc.

Developing hybrid mobile applications has its advantages. Since hybrid mobile applications take advantage of HTML5, CSS, and JavaScript web languages, there is usually a low learning curve for web developers. Also, hybrid mobile applications allow for one coded base for all platforms. The base code is simply wrapped in a native code

specific to the mobile platform where the application will be installed.

2.4 Going Hybrid

After the different methods for developing a mobile app were reviewed, it was determined that Walk Tracer would be developed as a hybrid mobile app. Walk Tracer was developed as hybrid mobile application for many different reasons. Walk Tracer was to be developed on the Android and iOS mobile platforms. That would make native development a very inefficient solution. If Walk Tracer had been developed as a native application for Android and iOS, then one would have had to learn two different programming languages. Additionally, future upgrades of Walk Tracer would have been difficult because one would have needed to write the upgrade twice in order for the app to be identical on both platforms. Also, if Walk Tracer had been developed as a mobile web application, then the app would have needed to be connected to the internet all times to function correctly. Taking those factors into consideration, hybrid development was determined to be the best development method for Walk Tracer.

2.4.1 PhoneGap

PhoneGap is a HTML5 application framework that is used to develop hybrid mobile applications [19]. It allows developers to program native applications by using web technologies. In other words, a developer can develop a smart phone or tablet application by using HTML, CSS, and JavaScript. PhoneGap is a JavaScript library that provides the JavaScript API access and control a mobile device's features like the camera or GPS.[19] The PhoneGap framework also has a native component which works in the background and does the actual work on the device.

Mobile applications that are developed using the PhoneGap framework have two parts: the JavaScript Business Logic Part and the JavaScript Part. The JavaScript

Business Logic Part is the part of the PhoneGap framework which controls the user interface and its functionality. The JavaScript Part is the part of the PhoneGap framework which accesses and controls the mobile device. *Figure 1* below shows the PhoneGap application architecture.

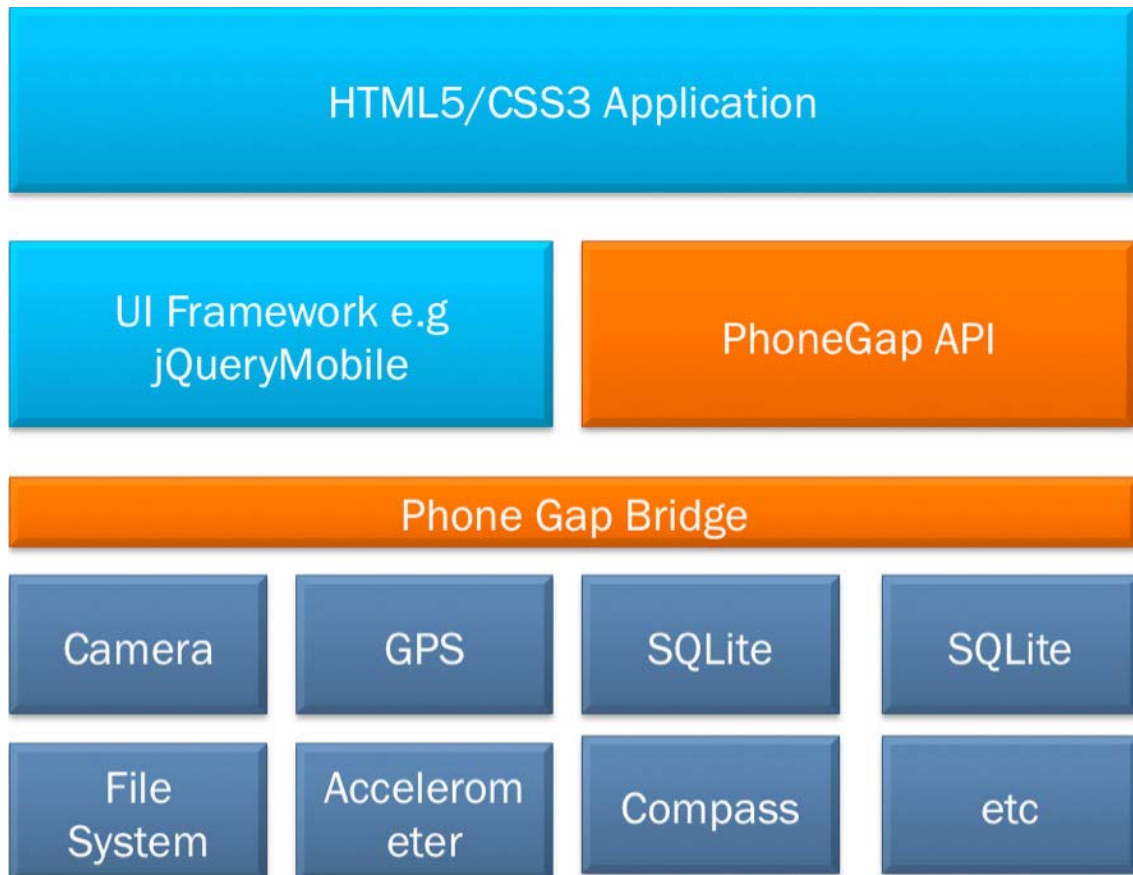


Figure 1 PhoneGap application architecture [19]

2.5 Naming the Application

Before the mobile application that was developed was given the official name of “Walk Tracer”, the names of several applications were reviewed. The names of these different mobile health applications compared to each other in order to evaluate similarities in their names. The research on the names of other mobile health applications showed that there were patterns in the names. Also the names of other mobile health apps

had the following similar characteristics:

1. contained four syllables or less.
2. closely matched the intended purpose of the app.
3. consisted of two to three words.

By taking these characteristics into account the name Walk Tracer was announced.

Chapter 3: Methodology

3.1 Overview

This chapter describes the development of the Walk Tracer application. Each page of the application was developed using the Waterfall methodology. Waterfall development consists of five different phases: requirements analysis, requirements definition, design, coding, testing, and maintenance [26]. Figure 2 shows a model of the waterfall methodology. Using this methodology will ensure that all requirements were met as well as ensure the quality of the app.

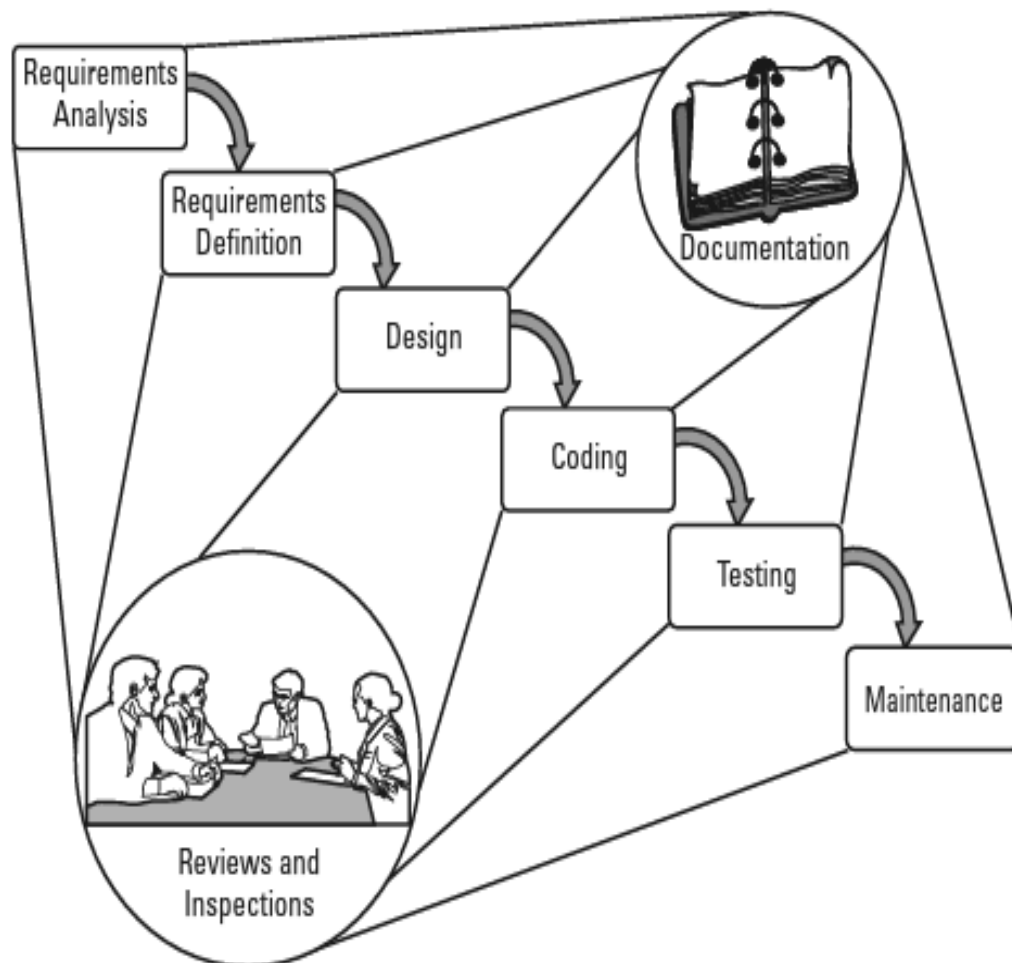


Figure 2 Waterfall Method

3.2 Use Case Diagram

A use case is a description of a sequence of actions to show a completed task or transaction in an application [21]. Figure 3 shows a use case diagram, which is a visually appealing graph of the user's interaction with the system. Descriptions of each use case are provided in Appendix A.

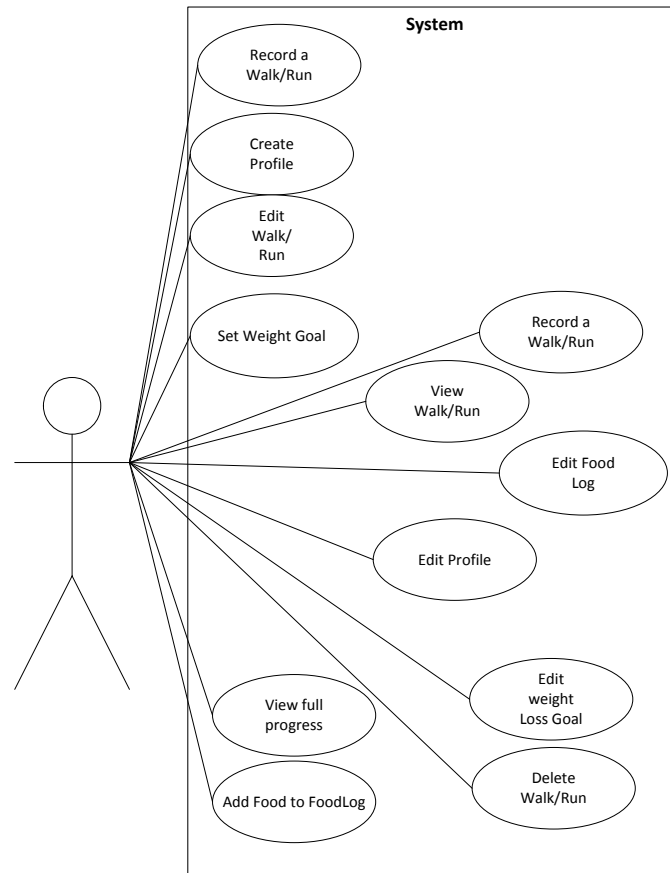


Figure 3 Use Case Diagram

3.3 User Interface Prototype

FluidUI was used to create a mockup of the future user interface Walk Tracer. FluidUI is a web application that allows the user to create prototypes of mobile applications. This prototype was created in order to help the client, as well as the

developer, get an initial idea of how the application would look and flow. In other words, this prototype was just a rough draft of Walk Tracer. This prototype is displayed in

Figure 4.

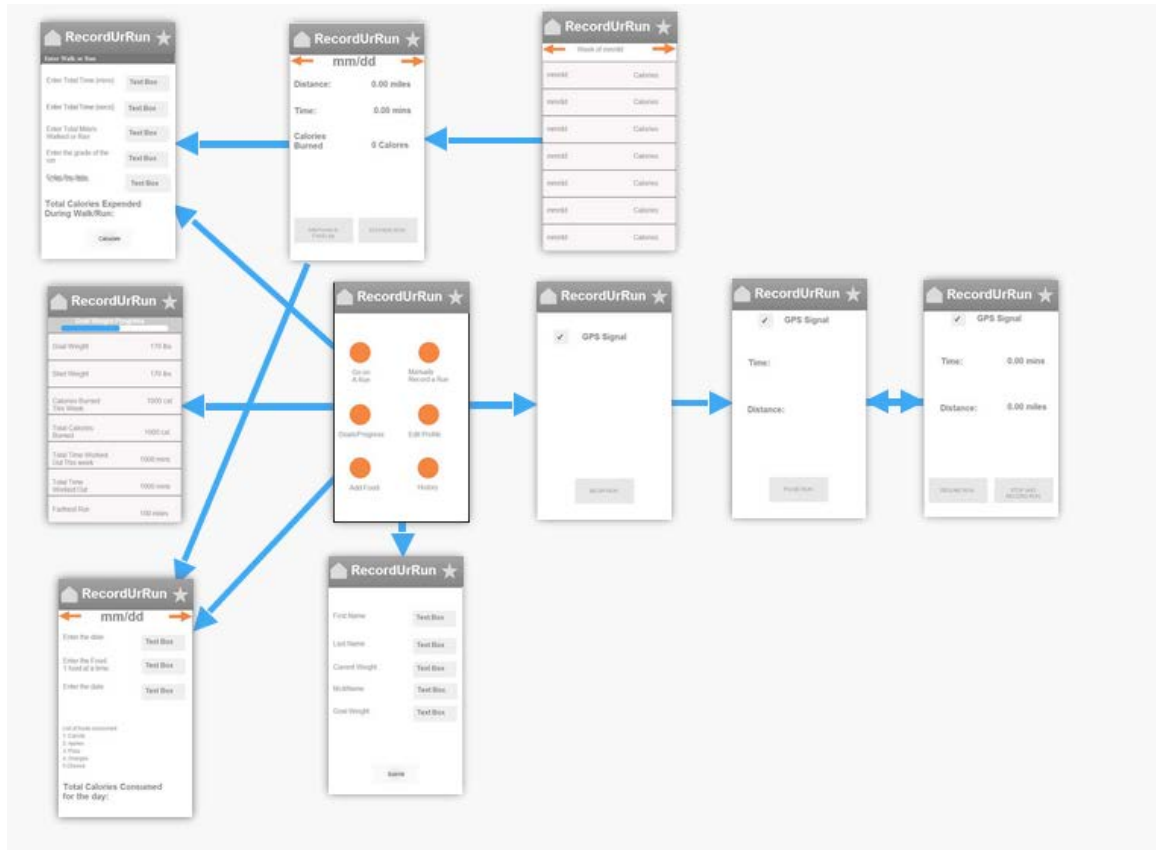


Figure 4 Early Page Prototype

3.3.1 Home Screen Page

The left screen in *Figure 5* shows the home screen of the Walk Tracer. The icons on the home screen reflect all the different features of the app. This is similar to the MapMyRun home screen that is shown on the right in *Figure 5*. This gives the user quick access to all features of the app. The final version of Walk Tracer app replaced the orange circles displayed in *Figure 5* with icons that reflect the page that the user is sent to. For example, the icon “Go On A Run” was replaced by the silhouette of a man running.

Pushing the home icon on the header is a quick link to the home screen. The home screen gives the user access to every feature of the developed mobile application.

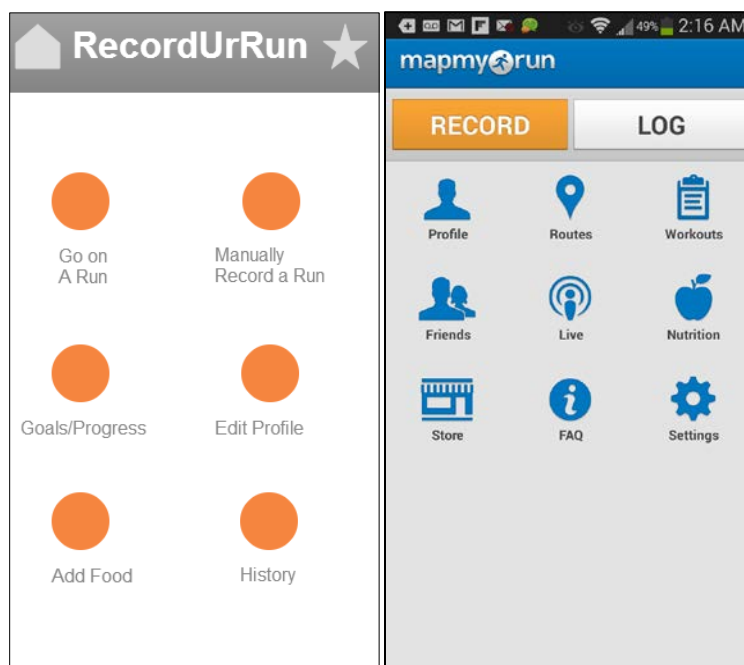


Figure 5-Home Screen of Prototype and MapMyRun Home Screen

3.3.2 Enter A Walk or Run Page

Another feature of Walk Tracer is that users are able to track their run using the mobile device's GPS. *Figure 6* shows three screens that the user sees while using this feature. This feature uses GPS to automatically track the users' distance as they run as well as calculate their time and speed. Through this screen the user will be able to start a run, pause/resume a run, and stop/ record a run. Walk Tracer can also calculate the current number of calories expended while the user is running. This feature is intended to be used during a user's run.

The GPS functionality of the phone is used by using the PhoneGap framework. Included in the PhoneGap framework is a "geolocation object that provides access to location data based on the devices GPS sensor or inferred from network signals" [20]. Geolocation has several methods that can be used to track a user's position over a given

period of time. There is a `GetCurrentPosition` method which was used to retrieve the user's current location. Next the `WatchPosition` method was used to track a user as they ran [20]. The geolocation methods provided the longitude and latitude of the device which was used to calculate the distance and speed of a user.

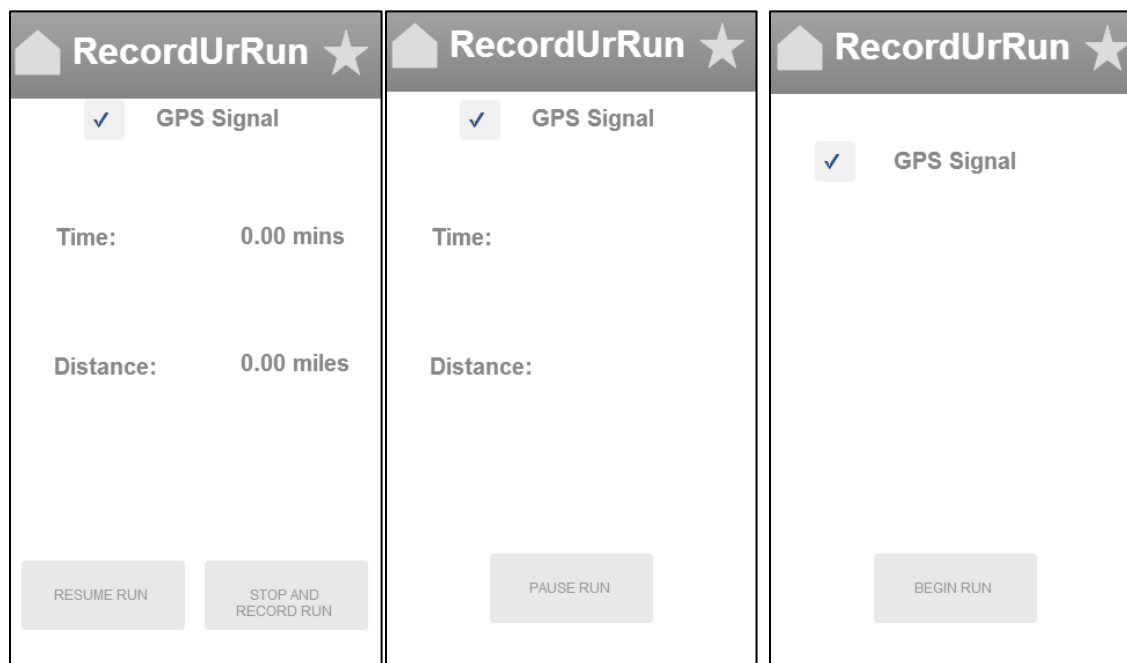


Figure 6 - Prototype GPS Run Entry

Additionally, a user is able to manually input a walk or run. This feature was provided just in case a user goes on a walk or run without his/her mobile device but still wanted to utilize the features of Walk Tracer. *Figure 7* shows the prototype user interface of the page that was used for a manual entry of a walk run. This page was also the page that a user saw if he/she chooses to edit a walk or run.

The screenshot shows the 'RecordUrRun' app interface. At the top, there is a header with a house icon, the text 'RecordUrRun', and a star icon. Below the header is a sub-header 'Enter Walk or Run'. The main content area contains five input fields, each with a 'Text Box' button to its right:

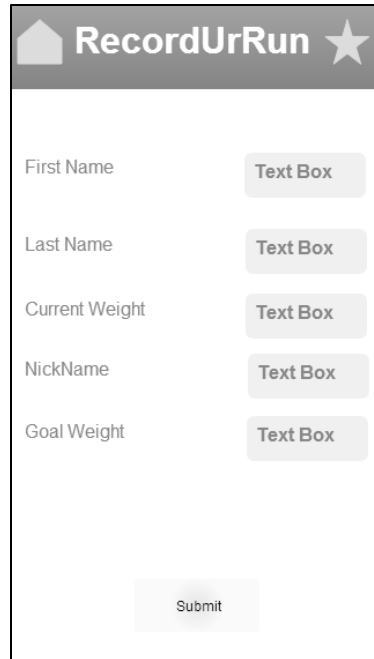
- Enter Total Time (mins) Text Box
- Enter Total Time (secs) Text Box
- Enter Total Mile/s Walked or Ran Text Box
- Enter the grade of the run Text Box
- Enter the date Text Box

Below these fields is a section titled 'Total Calories Expended During Walk/Run:'. At the bottom of this section are two buttons: 'Calculate' and 'Clear this Run'.

Figure 7 Manual Entry of Walk/Run

3.3.3 Profile Page

Walk Tracer is able to track a user's weight loss progression. In order to do this, an initial weight is obtained when the user first uses Walk Tracer. This information is gathered on the profile page. The profile page also records the user's first, last and nickname. The user is able to set a goal weight that he/she would like to reach. The user is expected to update his or her weight as they he/she loses weight while running and walking. An accurate weight is extremely important because current weight is used in the formulation to calculate the number of calories burned during a walk or run. An example of the profile page is shown in *Figure 8*.



The image shows a mobile application interface for 'RecordUrRun'. At the top, there is a dark grey header bar containing a home icon on the left, the text 'RecordUrRun' in the center, and a star icon on the right. Below the header, the main content area is white and contains five text input fields arranged vertically. Each field is labeled on the left and has a 'Text Box' button on the right. The labels are 'First Name', 'Last Name', 'Current Weight', 'NickName', and 'Goal Weight'. At the bottom center of the page, there is a 'Submit' button.

Figure 8 Profile Page

3.3.3 Food Log Page

Walk Tracer allows the users to keep track of the foods they eat by logging them in the app. The user specifies how many calories are in the food that they are entering. The food log page keeps a running total of calories consumed in a given day. The user will have the ability to clear the food log for a given day. *Figure 9* shows the user interface prototype for the Food Log page. The user is able to get to this page from the home screen as well as through weekly progress view. This feature was removed after a client meeting. The Food log page was considered out of scope.

Figure 9 Food Log Page

3.3.3 Progress Page

The application also keeps track of certain statistics, such as the farthest distance run, the total number of calories burned while using the app, the total number of calories burned for the current week, and the total number of minutes worked out while using the application. This page allows the user to track their overall progress as well as encourages the user to constantly grow by trying to beat their past achievements. The picture on the left of *Figure 10* demonstrates how the progress page will possibly look. The picture on the right of *Figure 10* shows the MyFitnessPal progress page. The MyFitnessPal progress page was used as inspiration for RecordUrRun's progress page.



Figure 10-View Progress Prototype Screen left, MyFitnessPal Progress Screen right

In addition to an overall progress view, RecordUrRun also has a weekly progress view. The screen on the left of Figure 10 shows what the weekly progress will look like. On the weekly progress page, seven days of workouts will be displayed, where each day will show the calorie deficit for that day. The calorie deficit for a day will be equal to the calories consumed minus the calories burned during a walk or run. If a user selects a specific day then they will be taken to the page on the right. The page on the right shows the specific details of a walk or run that day, caloric deficit, and also gives the user access to the Foods Log Page and the Edit/Add Run Page.

RecordUrRun		RecordUrRun	
← Week of mm/dd →		← mm/dd →	
mm/dd	Calories	Distance:	0.00 miles
mm/dd	Calories	Time:	0.00 mins
mm/dd	Calories	Calories Burned	0 Calores
mm/dd	Calories		
mm/dd	Calories		
mm/dd	Calories		
mm/dd	Calories		
mm/dd	Calories		
		Add Foods to Food Log	EDIT/ADD RUN

Figure 11 Weekly Progress Report Page

3.4 Changes to the Prototype

The prototype was shown to the client and the client provided feedback in order to improve Walk Tracer. As mentioned in an earlier section the food log was removed from the requirements. It could be a feature that is added to Walk Tracer in the future but the client wanted to focus on the caloric expenditure portion of the mobile application. The implementation of Google maps were added to the requirements.

3.5 Technologies

3.5.1 JQuery Mobile

A technology used in the development of Walk Tracer was jQuery Mobile. jQuery Mobile is a user interface (UI) framework that builds cross-platform mobile web applications[23]. It can also be compiled with hybrid technologies so that it can be distributed in mobile native app stores. This allows a developer to create applications that can run on almost every phone, tablet and desktop which grants a unified experience for all consumers [23].

jQuery Mobile was used mostly for the user interface of the app. By using this technology webpages are created a user interface that met HTML5 and CSS3 standards. In addition to meeting web standards, the created user interface could be implemented on any mobile device. jQuery Mobile has the ability to easily make changes to the style and color of the UI.

3.5.2 JavaScript

Another technology used in the development of Walk Tracker was JavaScript. JavaScript is described as a web enhancing technology and is used to turn static webpages into interactive and engaging webpages. JavaScript is a tool that can be used to get a webpage to respond or react directly to user interaction with form elements such as input fields, text areas, buttons and checkboxes[24]. It can work side-by-side and enhance the HTML5 and CSS3 technologies.

3.6 Software Evaluation and Testing

RecordUrRun needed to be a high quality and working mobile application, which required sufficient testing. As stated at the beginning of this chapter, each page of the mobile application was developed using the waterfall method. One phase of the waterfall method is testing. This means that each page of the mobile application was tested through unit testing in order to ensure that the page is functional. Upon the completion of RecordUrRun development, the application was as a whole application.

3.6.1 Testing procedures

In order to ensure that the application works properly, two different routes around were mapped out in the city. The routes each had a different distance and location. Two test users were created and the application was installed on two different android devices. Each test user's walk/run for each route was taken using the GPS feature. Also,

walk/runs were manually entered using the distances from the previously selected routes.

Chapter 4: Outline of Completed Project

4.1 Setting up the environment

In order to develop the application the Microsoft Visual Studio 2013 integrated development environment (IDE) was installed. The main reason Microsoft Visual Studio was is because it has a multi-device hybrid add-on extension. This add-on included an android emulator which allows one to get a realistic view of how the application would look on an android device. Originally, the use of the Visual Studios was not planned in the development in of Walk Tracer. However, the inclusion of the multi-device hybrid add-on allowed for additional testing that was not originally planned.

4.2 Getting Started-Homepage

The first page developed was the home screen; the page to which the user is initially taken when beginning the exercises with the Walk Tracer. As stated in the previous section, the homepage consists of several icons that take the user to the various pages. The initial prototype of Walk Tracer neglected to include icons, however, and that pointed to the possible need for a graphic designer to be included in planning the specifications for an application under development. Alternatively, Adobe Illustrator was used as a less expensive method of creating simple icons that could be used on the homepage.

4.3 Profile page

The profile page is very important because it is the foundation for all the other pages of the app. All the caloric equations use the current weight to calculate caloric expenditure. If the user does not have a current weight entered into the app then the caloric equations will be inaccurate. Once the submit button is pushed, a jquery on click event is called and the user's information is stored using the browser's local storage.

Local storage stores data using a unique key. Table 3 shows the unique keys that were used to store the user's profile information. The goal when choosing names for keys was to use a name that would give a future developer an idea of the type of data that the key stored.

Table 3 Local Storage Keys

Key	Description
user.firstname	Stores the user's first name
user.lastname	Stores the user's last name
user.currentWeightPounds	Stores the user's current weight
user.goalWeightPounds	Stores the user's goal weight
user.currentWeightKG	The caloric expenditure formula uses kilograms instead of pounds. I divided the current weight by 2.2 and stored the result in local storage.

4.4 *Manual Run and Handling Local Storage*

The development of the user interface of the manual run page was fairly simple. It just consisted of text boxes and a submit button at the bottom. The most challenging part was figuring out how to implement local storage on this page. This page needed the ability to calculate the calories used, inform the user of the calories used, and store specific details about a walk or run. Also, anything placed in the local storage had to be retrievable.

In the previous section, it was explained that the data stored in local storage must be stored with a unique key. This unique key is used to retrieve the data at a later time. The unique key used was the date that the user inputted. Using the date as the unique key meant that all the dates that the user used had to be in a specific format. The user was

forced to input the date in the required format by using data validation techniques which is discussed in a future section of this paper.

When the user clicks submit the current weight is pulled from local storage and used to calculate the calories used during the walk or run. After the calories used are calculated that amount is displayed to the user and then the details of the run are stored in an array that is pushed into local storage. The array is linked to the unique ID which is the date of the run. The details of the run that are stored are duration, distance and calories used.

4.5 The Road to Data validation

As Walk Tracer was being developed, it was discovered that a user had numerous chances to break the app. For example, if alphabetical text was placed into a text box that required only numerical input then my Walk Tracer would get an error once it tried to use the text in the caloric expenditure in equation. Another error that would occur is if a user attempted to use the walk or run entry pages without having to set up their profile. In order to alleviate these potential problems data validation and error checking had to be added to the app.

In order to prevent a user from submitting text JavaScript alerts and if statements were used. Alerts are JavaScript events that pop up a message on a user's screen when called. Figure 12 shows an example of a JavaScript alert. Alerts were programmed to fire, once the user pushed submit, if the user inputted incorrect data into the profile page. The alert would inform the user of their error and how to fix it. Although, alerts worked really well, other options for data validation were explored. Also, some browsers allow you to turn off alerts and if that were the case then there would be no error checking.

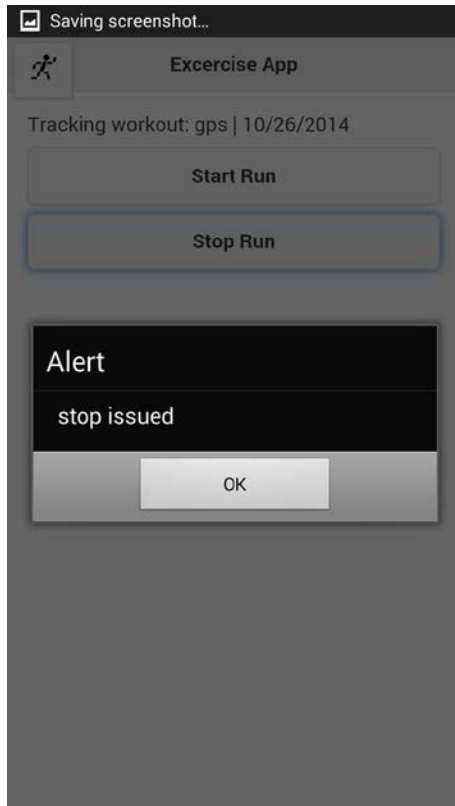


Figure 13 Showing off an alert from Javascript

In an effort to prevent possible errors another route of data validation was used instead of alerts. HTML5 provides an attribute to text boxes that allows the user to specify what input type expected to be typed into the text box. Some of the categories include number, date, color, time, email and many others. JQuery mobile uses input type to specify what keyboard is displayed when the user clicks the text box. Figure 13 shows some examples of the keyboards that android will display based upon the input type. The advantage to using input type is that the user is limited as to what he or she can put into a text box.

After adding the different input types to all my text boxes, it was tested on an android device. The input field worked perfectly for standard keyboards and date inputs, but the user could switch back to the standard keyboard from the numerical keyboard. This meant that the user could input alphabetical text into a number field which would

cause an error in the mobile application. After some research was conducted, jQuery Mobile's keypress function was found. The keypress function has the ability to run some code every time the user pushes a key on the keyboard. The keypress function was used to check if the key that the user pressed was a number. If the key that the user pressed was not a number then the app simply ignores the key and will not update the text box with the user's selection. This prevents the user from inputting any invalid text into the number field.

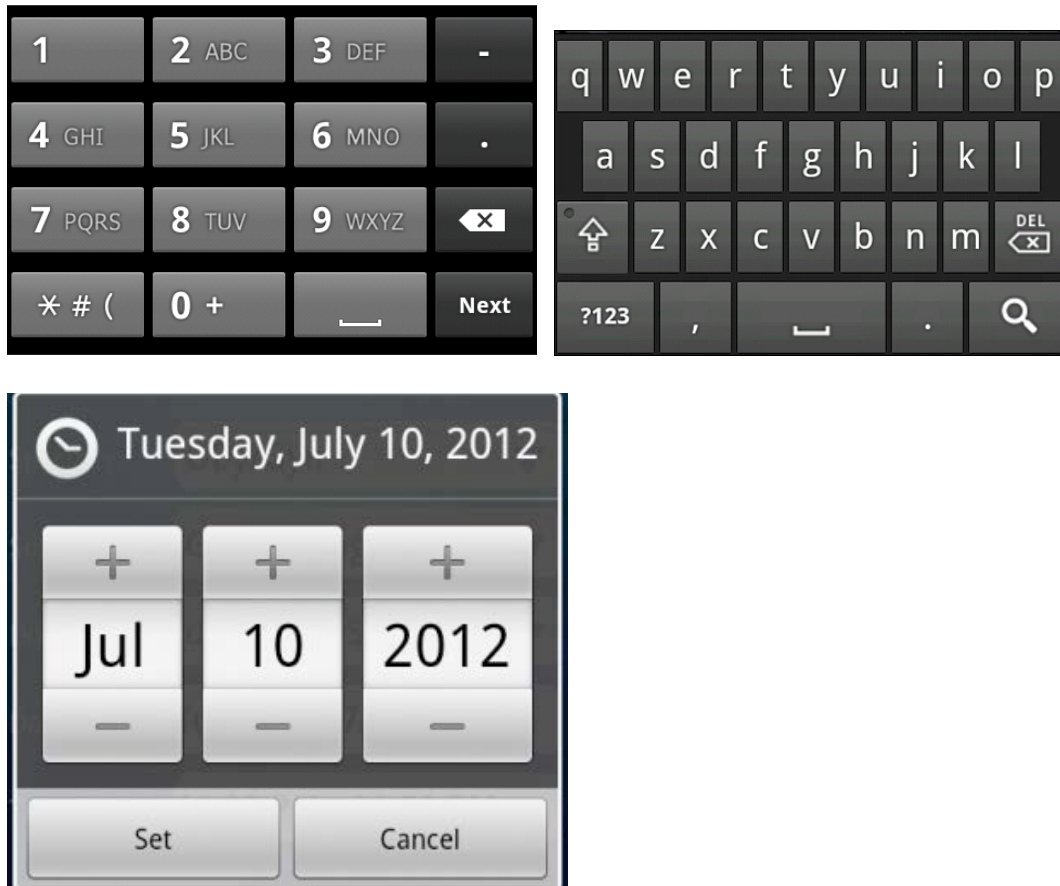


Figure 14 Android Screen shot of number keyboard, standard keyboard, and date picker in order

4.6 PhoneGap Geolocation

The next page that was developed was the GPS run/walk page. As mentioned in the previous section, this page provided the user the ability to track his/her run by using

the phone's navigation feature. Phonegap's API provided access to the device's geolocation capabilities. The page was coded by implementing Phonegap's "geolocation.getCurrentPosition" function when a user started the run, by pushing the start run button. Phonegap's "getCurrentPosition" function grabs the user's current location through several different "position" attributes such as latitude, longitude, altitude and timestamp. Next Phonegap's "watchPosition" function, which tracks the user as they move positions, was used. The user's position is updated every few seconds and his/her new "position" attributes are placed in an array and stored in local storage. The user's position is tracked until the "stop run" button is pushed.

After all the tracking data for a user was accumulated, the next step was to determine how to calculate the distance that the user had traveled with the tracked data. The haversine's formula was used to calculate the distance the user had ran/walked[22]. "Haversine's formula is used to calculate the great-circle distance, the shortest distance over the earth's surface, between two points"[25]. Haversine's formula calculates distance in kilometers so the distance had to be converted into miles, which was needed for ACSM's caloric expenditure equation. Next the distance that the user ran/walked is placed in local storage. All this occurs once the "stop run" button is pressed.

4.6.1 Adding Google Maps

During the proposal, it was suggested that a maps feature be added to the application. Research was conducted on exactly how to get this accomplished. The Google maps API ended up being used to complete this task. In order to use this API, one had to register with Google maps so that he/she could receive a unique key that would identify him/her as a user of the Google API.

The next step was to figure out how Google Maps would be implemented into Walk Tracer. After much consideration, it was decided that a user should be able to look up his/her walk or run and see the route that he/she took. A button called “see my route” was added to Walk Tracer. The “see my route” button allows a user; if they used GPS to record his or her walk/run, to see the route they took the Google maps.

4.7 History and Records Pages

The history and the records pages were developed to give the user the ability to track their progression while using the application. The data for those pages are supplied by using local storage. In reference to the history pages, daily and weekly, the user’s run/walk data for specific date is displayed when the user selects that date. The records page updates itself in local storage as new milestones are reached. For example, on the records page there is a category called longest run. If a user currently inputs a run in Walk Tracer that is longer than their current longest run, then the category longest run is replaced by the current run.

Chapter 5: Conclusion

5.1 *Project Deliverables*

Listed below are the deliverables for the project:

- 1) Final defense paper
- 2) A fully operational and client approved Walk Tracer Android mobile application.
 - a. Application should be ready for release on the android market.
- 3) Well-documented code so that another developer could easily catch up and add additional features to the application.
- 4) A page by page user guide on how to use Walk Tracker.

5.2 *Teaching/Learning Aspects of this Development Process*

One of the most important aspects in beginning the development of a new application may be to research the literature of other applications to understand their types, composition, and functionality. The developer can learn a great deal from the products that are already on the market. The three types already in use, hybrid, web app, and native, all have pros and cons that challenge the developer to choose between them so that individual needs are met.

Additionally a visually appealing application may require the assistance of a specialist in certain fields in order to achieve. Since developers, desirous of satisfying clients or providing marketable and competitive products, may lack certain skills in making a visually appealing object it may be wise to solicit the assistance of paid professionals who possess these needed skills.

The most advantageous of the lessons learned during a development process may be use of programming languages. Despite the fact that books are available to acquaint the user with the existence of that language, its application in an actual project is entirely different than simply reading about it. This project required the use of JavaScript and jQuery programming languages, and the best way of learning these languages occurred during the actual practice of programming the languages into the project. The amount of troubleshooting required may cause a great deal of anxiety, but the learning of these languages make it well worth the effort.

5.3 *Future Work*

The steps listed below should be followed for the successful design of the Walk Tracer.

- 1) Do beta testing with android users.
- 2) Hire a graphic designer to design logos for Walk Tracer.
- 3) Add a food database.
 - a. The food database is found in many applications according to the literature.
 - b. At some point the food database will probably become a part of the Walk Tracer.
- 4) Add a cloud service and back of user's information online and eventually make a website that is linked to the mobile app.
- 5) Make the app available for all platforms.

References

- [1] Thompson, Walter R., Neil F. Gordon, and Linda S. Pescatello. *ACSM's Guidelines for Exercise Testing and Prescription*. Philadelphia: Lippincott Williams & Wilkins, 2010. Print. 158
- [2] "What Is Vo2max?." Northern Territory News (Includes Sunday Territorian) (n.d.): Newspaper Source Plus. Web. 4 Nov. 2013.
- [3] <http://searchsoftwarequality.techtarget.com/definition/native-application-native-app>
- [4] <http://www.webopedia.com/TERM/C/CSS.html>
- [5] Savitz E, Quilligan A. HTML5 Vs. Native Mobile Apps: Myths and Misconceptions. Forbes.Com [serial on the Internet]. (2013, Jan 23), [cited January 20, 2014]; 27. Available from: Business Source Complete.
- [6] Budiu, Raluca. "Mobile: Native Apps, Web Apps, and Hybrid Apps." *Nielsen Norman Group*. N.p., 09 14 2013. Web. 20 Jan 2014.
- [7] Korf, M., and E. Oksman. N.p., n. d. 27 Jan 2014.
<http://wiki.developerforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options>.
- [8] "American College of Sports Medicine." American College of Sports Medicine. 02 Feb. 2014 <<http://www.acsm.org/>>.
- [9] Comstock, Johan. "Pew: 19 percent of smartphone users have health apps." *Mobihealthnews.com*. 12 Nov. 2012. 02 Feb. 2014
<<http://mobihealthnews.com/18965/pew-19-percent-of-smartphone-users-have-health-apps/>>.
- [10] Smith, Aaron. "Smartphone Ownership 2013." 56% of American adults are now smartphone owners. 5 June 2013. 02 Feb. 2014
<<http://pewinternet.org/Reports/2013/Smartphone-Ownership-2013/Findings.aspx?view=all>>.
- [11] <http://www.nngroup.com/articles/mobile-native-apps/>
- [12] <http://www.hswsolutions.com/services/mobile-web-development/mobile-website-vs-apps/>
- [13] Anglin, Todd. "Approaching Mobile: Understanding the Three Ways to Build Mobile Apps." *www.telerik.com*. Telerik. 4 April 2014.
- [14] "About Runtastic | Runtastic - makes sports funtastic." Runtastic GmbH 2014, n.d. Web. 5 May 2014. <<http://www.runtastic.com/en/about>>.

- [15] "Calorie Counter for Android Overview." The Best Calorie Counter and Food Diary App for Android. MyNetDiary Inc., n.d. Web. 5 May 2014. <<http://www.mynetdiary.com/android-calorie-counter.html>>.
- [16] "MapMyRun." *Running, Running Maps and Running Routes, Runners Community*. MapMyFitness, Inc., n.d. Web. 5 May 2014. <<http://www.mapmyrun.com/>>.
- [17] "RunKeeper ." *RunKeeper*. FitnessKeeper, 2014, n.d. Web. 5 May 2014. <<http://runkeeper.com/>>.
- [18] Dolan, Brian. "13K iPhone consumer health apps in 2012." *mobihealthnews RSS*. MobiHealthNews, 22 Sept. 2011. Web. 6 May 2014. <<http://mobihealthnews.com/13368/report-13k-iphone-consumer-health-apps-in-2012/>>.
- [19] Ghato, Rohit, and Yogesh Patel. "Understanding Cross-Platform Mobile Application Development." *Beginning PhoneGap mobile Web framework for JavaScript and HTML5*. New York: Apress, 2012. . Print.
- [20] "PhoneGap API Documentation." . N.p., n.d. Web. 12 May 2014. <<http://docs.phonegap.com/en/3.4.0/index.html>>.
- [21] Dobing, Brian, and Jeffrey Parsons. "Understanding the Role of use Cases in UML: A Review and Research Agenda." *Journal of Database Management* 11.4 (2000): 28-36. *ProQuest*. Web. 1 Sep. 2014.
- [22] Robusto C. 1957. The cosine-haversine formula. *Am. Math. Monthly* 64: 38–40
- [23] Broulik, Brad. *Pro JQuery Mobile*. Berkeley, CA: Apress, 2011. Internet resource.
- [24] Goodman, Danny, and Michael Morrison. *Javascript Bible*. Indianapolis: John Wiley, 2004. eBook Collection (EBSCOhost). Web. 4 Nov. 2014.
- [25] Veness, Chris. "Calculate Distance and Bearing between Two Latitude/Longitude Points Using Haversine Formula in JavaScript." *Movable Type Scripts*. Web. 5 Nov. 2014.
- [26] Rakitin, Steven R. *Software Verification And Validation For Practitioners And Managers*. Boston: Artech House, 2001. eBook Collection (EBSCOhost). Web. 5 Nov. 2014.
- [27] "Diet Or Exercise, It's Calories That Count." Tufts University Health & Nutrition Letter 25.2 (2007): 1. SPORTDiscus with Full Text. Web. 5 Nov. 2014.

Appendix A
Walk Tracer App Use Cases

Appendix A: Use Cases

Use Case Name	Edit Walk/Run	
Scenario	User Selects Edit Walk/Run	
Brief Description	A user has a walk or run that they would like to edit.	
Actors	User	
Related Use Cases	Delete Walk/Run, Enter Walk/Run	
Preconditions	In order to edit a walk or run a user must have already recorded a walk or run	
Postconditions	The walk/run is deleted and any calculations using the walk/run are deleted as well. For example the user's totals calories lost.	
Flow of Events	Actor	System
	1. The user selects the date of the walk/run that they want to edit.	
		2. The app displays information from the previous walk/run
	3. The user pushes edit.	
		4. The system changes the labels to editable text fields.
	5. The user makes changes and saves those changes.	
		6. The system saves those changes in local storage.
Exception Conditions	1. The user does not have a walk or run recorded for the selected date. 1a. The user will not receive a response when they click the date of the walk/run that they wish to edit.	

Use Case Name	View user history (Daily)	
Scenario	User Selects History	
Brief Description	A user wants to view the daily progress they have made while using the app.	
Actors	User	
Related Use Cases	View Weekly History	
Preconditions	A user must have a profile in order to track progress and record walk/run data in order for results to be displayed	
Postconditions	A users data will be displayed on the screen.	
Flow of Events	Actor	System
	1. The user selects view daily history	
		2. The retrieves and displays the current date.
		3. The app goes into local storage and pull the data for selected date
		4. The app displays the data on the user's screen
		5a. If there is no run data for that day the "add a run" button will be displayed.
		5b. If there is run data then the "edit run" button will be displayed
	6. The html5 elements are updated with the data from local storage	
Exception Conditions	1. The user does not have a walker one recorded for the selected date. 1a. The system will put in zero's for all the data fields	

Use Case Name	Record Walk/Run (Manually Entry)	
Scenario	User Selects Record Walk/Run	
Brief Description	A user would like to enter a walk or run	
Actors	User	
Related Use Cases	Delete Walk/Run, Edit Walk/Run	
Preconditions	The user has completed a walk or run without GPS	
Postconditions	If the user has entered a weight then the calories burned will be calculated for the walk or run	
Flow of Events	Actor	System
	1. The user selects the date of the walk/run they want to enter or they select enter walk/run on the home page.	
		2. The app will take them to the enter walk/run page
	3. The user will enter the required information for a walk or run and click submit	
		4. The system will calculate the calories that the user lost during the walk/run
		5. The system will save the details of the walk or run info in local storage.
		6. The app will display a summary of the walk/run to the user
	7. The user will see the displayed information	
Exception Conditions	<ol style="list-style-type: none"> 1. The use does not input all the required information <ol style="list-style-type: none"> a. The app will display an error. b. The app will stay on the Record a Run page 	

Use Case Name	Update the user profile	
Scenario	The user clicks the profile icon	
Brief Description	A user wants to update their profile	
Actors	User	
Related Use Cases	Create user profile	
Preconditions	A user must have created a user profile to be able to update it.	
Postconditions	The profile is update with the new user data and all future calculations will be calculated using that data	
Flow of Events	Actor	System
	1. The user goes to the profile page	
		2. The system will check in local storage and display the current user information
	3. The user will input their information and click submit	
		4. The system will replace the user's data in local storage
Exception Conditions	1. The use does not input all the required information <ul style="list-style-type: none"> a. The app will display an error. 	

Use Case Name	View user history (Weekly)	
Scenario	User Selects History	
Brief Description	A user wants to view the weekly progress they have made while using the app.	
Actors	User	
Related Use Cases	View Daily History	
Preconditions	A user must have a profile in order to track progress and record walk/run data in order for results to be displayed	
Postconditions	A user's data for the week will be displayed on the screen.	
Flow of Events	Actor	System
	1. The user selects view weekly history	
		2. The retrieves and displays the current date.
		3. The app goes into local storage and pull the data for selected date
		4. The app displays the data on the user's screen
		5a. If there is no run data for that day the "add a run" button will be displayed.
		5b. If there is run data then the "edit run" button will be displayed
	6. The html5 elements are updated with the data from local storage	
Exception Conditions	1. The user does not have a walker one recorded for the selected week. 1a. The system will put in zero's for all the data fields	

Use Case Name	View user records/goals	
Scenario	User selects view progress	
Brief Description	A user wants to view the progress they have made while using the app.	
Actors	User	
Related Use Cases	View Daily/ Weekly History, Delete user records / goals	
Preconditions	A user must have a profile in order to track progress	
Postconditions	A users records/goals will be displayed on the page	
Flow of Events	Actor	System
	1. The user selects view records	
		2. The app goes into local storage and pulls all the users data out of local storage
		3.The system will put the data on the various elements on the html5 page
Exception Conditions	1. If the user does not have a profile 1a. The user will be redirected to the profile page	

Use Case Name	Delete Walk/Run	
Scenario	User selects delete on the walk/run	
Brief Description	A user wants to delete a previous walk/run they have made	
Actors	User	
Related Use Cases	Add Walk/Run, Edit Walk/Run	
Preconditions	A user must have a walk/ run to delete	
Postconditions	A user's walk/run will be deleted from the system and removed from local storage.	
Flow of Events	Actor	System
	1. The user goes to the daily run page	
	2. The user selects delete run	
		3. System will issue a confirmation
	4. The user will confirm the delete	
		5. System will look up the walk/run in local storage
		6. System will delete the walk/run from local storage
		7. System will issue a delete confirmation
Exception Conditions		

Use Case Name	Delete all App Data	
Scenario	User is on profile page and selected delete data	
Brief Description	User wished to delete all their app data	
Actors	User	
Related Use Cases	Delete Run for specific date	
Preconditions		
Postconditions	A user will not have any info in the app	
Flow of Events	Actor	System
	1. The user selects delete all data button	
		2. The system will issue a confirmation.
	3. User accepts confirmation	
		4. System deletes all info in local storage
		5. System issues confirmation
Exception Conditions		

Use Case Name	View Map of Run	
Scenario	The user selects view the map	
Brief Description	The user wants to see the route they ran	
Actors	User	
Related Use Cases	View my progress daily	
Preconditions	A user must have ran using the GPS feature	
Postconditions	The user can see the route they walked or ran	
Flow of Events	Actor	System
	1. The user selects view my map	
		2. The app gets the date for the day the user selected
		3. The system goes in to local storage and makes sure that a GPS run is present
		4. The system pulls all of the longitudes and latitudes for a run
		5. The system displays the google map
		6. The system plots the longitude and latitudes on the google map
	7. The system displays the google map with the plots	
Exception Conditions	1. The user does not have a GPS walker one recorded for the selected date. 1a. The map will not display	

Appendix B

Walk Tracer Project Schedule

Appendix B: Project Schedule

Task Name	Duration	Start	Finish	Predecessors
Record Ur Run Development	173 days	Mon 3/3/14	Wed 10/29/14	
Proposal	133 days	Mon 3/3/14	Wed 9/3/14	
Develop 1st Draft of Proposal Document	58 days	Mon 3/3/14	Wed 5/21/14	
Develop 2nd Draft of Proposal Document	69 days	Tue 5/27/14	Fri 8/29/14	2
Defend Proposal	1 day	Wed 9/3/14	Wed 9/3/14	3
Profile Page/Home Screen	5 days	Thu 9/4/14	Wed 9/10/14	1
Create Profile Page	1 day	Thu 9/4/14	Thu 9/4/14	
Edit Profile	2 days	Fri 9/5/14	Mon 9/8/14	6
Create Home Screen	2 days	Tue 9/9/14	Wed 9/10/14	7
Progress Page	2 days	Thu 9/11/14	Fri 9/12/14	5
Weekly Progress Page	1 day	Thu 9/11/14	Thu 9/11/14	
Daily Progress Page	1 day	Thu 9/11/14	Thu 9/11/14	
Insert Entry	1 day	Thu 9/11/14	Thu 9/11/14	
Delete Entry	1 day	Thu 9/11/14	Thu 9/11/14	
Testing	2 days	Thu 9/11/14	Fri 9/12/14	
Walk/Run Manual Entry	4 days	Mon 9/15/14	Thu 9/18/14	9
Review preliminary software specifications	1 day	Mon 9/15/14	Mon 9/15/14	9
Enter A Walk/Run Page	3 days	Tue 9/16/14	Thu 9/18/14	16
Clear Run	1 day	Tue 9/16/14	Tue 9/16/14	16
Testing	2 days	Wed 9/17/14	Thu 9/18/14	18
Walk/Run GPS Entry	11 days	Fri 9/19/14	Fri 10/3/14	15
Review functional specifications	1 day	Fri 9/19/14	Fri 9/19/14	
Development	5 days	Mon 9/22/14	Fri 9/26/14	21
Testing	5 days	Mon 9/29/14	Fri 10/3/14	22
Map 3 different routes around town	1 day	Mon 9/29/14	Mon 9/29/14	
Run Those Routes using App	2 days	Tue 9/30/14	Wed 10/1/14	24
Review Results	2 days	Thu 10/2/14	Fri 10/3/14	25
Food Log Page	5 days	Mon 10/6/14	Fri 10/10/14	20
Review preliminary software specifications	1 day	Mon 10/6/14	Mon 10/6/14	
Enter Food	2 days	Tue 10/7/14	Wed 10/8/14	28
Edit Food	2 days	Tue 10/7/14	Wed 10/8/14	28
Delete Food	2 days	Tue 10/7/14	Wed 10/8/14	28
Calculate Calories Burned	2 days	Tue 10/7/14	Wed 10/8/14	28
Testing	2 days	Thu 10/9/14	Fri 10/10/14	29,30,31,32
Testing	5 days	Mon 10/13/14	Fri 10/17/14	27
Build and Run Test Cases	5 days	Mon 10/13/14	Fri 10/17/14	
Defense	8 days	Mon 10/20/14	Wed 10/29/14	34
Final Draft for Capstone	5 days	Mon 10/20/14	Fri 10/24/14	
Defend	3 days	Mon 10/27/14	Wed 10/29/14	37

Appendix C

American College Association of Sports Medicine Formulas

ACSM metabolic equations

The following equations are what I used to calculate caloric expenditure for a walk/run

1)

If the user's mph is less than 5 I used this equation:

$$\text{VO}_2 \text{ (mL} \cdot \text{kg}^{-1} \cdot \text{min}^{-1}\text{)} = (0.1 \cdot \text{S}) + (1.8 \cdot \text{S} \cdot \text{G}) + 3.5 \text{ mL} \cdot \text{kg}^{-1} \cdot \text{min}^{-1}$$

[1]

If the user's mph was 5 mph or greater than I used this equation:

$$\text{VO}_2 \text{ (mL} \cdot \text{kg}^{-1} \cdot \text{min}^{-1}\text{)} = (0.2 \cdot \text{S}) + (0.9 \cdot \text{S} \cdot \text{G}) + 3.5 \text{ mL} \cdot \text{kg}^{-1} \cdot \text{min}^{-1}$$

[1]

2)

Multiply the VO_2 value in ml/kg/min by the user's weight in kilograms. [1]

3)

Divide this value by 1,000 to convert VO_2 to L/min [1]

4)

Liters of VO_2 x 5 kcals/liter O_2 = kcalories [1]

Below is an explanation of the various abbreviations [1]:

VO_2 = horizontal component + vertical component + resting component

VO_2 is gross oxygen consumption in $\text{mL} \cdot \text{kg}^{-1} \cdot \text{min}^{-1}$

S is speed in $\text{m} \cdot \text{min}^{-1}$

BM is body mass (kg)

G is the percent grade expressed as a fraction

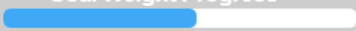
Work rate ($\text{kg} \cdot \text{m} \cdot \text{min}^{-1}$)

Appendix D
Project Prototype

<div style="background-color: #444; color: white; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> RecordUrRun ★ </div> <div style="text-align: center; margin-top: 10px;"> ← mm/dd → </div> <div style="margin-top: 20px;"> <p>Distance: 0.00 miles</p> <p>Time: 0.00 mins</p> <p>Calories Burned 0 Calores</p> </div> <div style="display: flex; justify-content: space-around; margin-top: 20px;"> <div style="border: 1px solid #ccc; padding: 5px; width: 40%;">Add Foods to Food Log</div> <div style="border: 1px solid #ccc; padding: 5px; width: 40%;">EDIT/ADD RUN</div> </div>	<div style="background-color: #444; color: white; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> RecordUrRun ★ </div> <div style="text-align: center; margin-top: 10px;"> ← Week of mm/dd → </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">mm/dd</td><td style="text-align: center;">Calories</td></tr> <tr><td style="text-align: center;">mm/dd</td><td style="text-align: center;">Calories</td></tr> <tr><td style="text-align: center;">mm/dd</td><td style="text-align: center;">Calories</td></tr> <tr><td style="text-align: center;">mm/dd</td><td style="text-align: center;">Calories</td></tr> <tr><td style="text-align: center;">mm/dd</td><td style="text-align: center;">Calories</td></tr> <tr><td style="text-align: center;">mm/dd</td><td style="text-align: center;">Calories</td></tr> <tr><td style="text-align: center;">mm/dd</td><td style="text-align: center;">Calories</td></tr> <tr><td style="text-align: center;">mm/dd</td><td style="text-align: center;">Calories</td></tr> </table>	mm/dd	Calories	mm/dd	Calories	mm/dd	Calories	mm/dd	Calories	mm/dd	Calories	mm/dd	Calories	mm/dd	Calories	mm/dd	Calories
mm/dd	Calories																
mm/dd	Calories																
mm/dd	Calories																
mm/dd	Calories																
mm/dd	Calories																
mm/dd	Calories																
mm/dd	Calories																
mm/dd	Calories																
<div style="background-color: #444; color: white; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> RecordUrRun ★ </div> <div style="margin-top: 20px;"> <p>First Name <input style="width: 80%;" type="text" value="Text Box"/></p> <p>Last Name <input style="width: 80%;" type="text" value="Text Box"/></p> <p>Current Weight <input style="width: 80%;" type="text" value="Text Box"/></p> <p>NickName <input style="width: 80%;" type="text" value="Text Box"/></p> <p>Goal Weight <input style="width: 80%;" type="text" value="Text Box"/></p> </div> <div style="text-align: center; margin-top: 20px;"> <div style="border: 1px solid #ccc; padding: 10px 20px; display: inline-block;">Submit</div> </div>	<div style="background-color: #444; color: white; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> RecordUrRun ★ </div> <div style="background-color: #444; color: white; padding: 5px; text-align: center; font-weight: bold; margin-top: 5px;">Enter Walk or Run</div> <div style="margin-top: 10px;"> <p>Enter Total Time (mins) <input style="width: 80%;" type="text" value="Text Box"/></p> <p>Enter Total Time (secs) <input style="width: 80%;" type="text" value="Text Box"/></p> <p>Enter Total Mile/s Walked or Ran <input style="width: 80%;" type="text" value="Text Box"/></p> <p>Enter the grade of the run <input style="width: 80%;" type="text" value="Text Box"/></p> <p>Enter the date <input style="width: 80%;" type="text" value="Text Box"/></p> </div> <div style="margin-top: 20px;"> <p>Total Calories Expended During Walk/Run:</p> </div> <div style="display: flex; justify-content: center; margin-top: 10px;"> <div style="border: 1px solid #ccc; padding: 10px 20px; display: inline-block; margin-right: 20px;">Calculate</div> <div style="border: 1px solid #ccc; padding: 10px 20px; display: inline-block;">Clear this Run</div> </div>																

RecordUrRun

Goal Weight Progress



Goal Weight	170 lbs
Start Weight	170 lbs
Calories Burned This Week	1000 cal
Total Calories Burned	1000 cal
Total Time Worked Out This week	1000 mins
Total Time Worked Out	1000 mins
Farthest Run	100 miles

RecordUrRun

GPS Signal

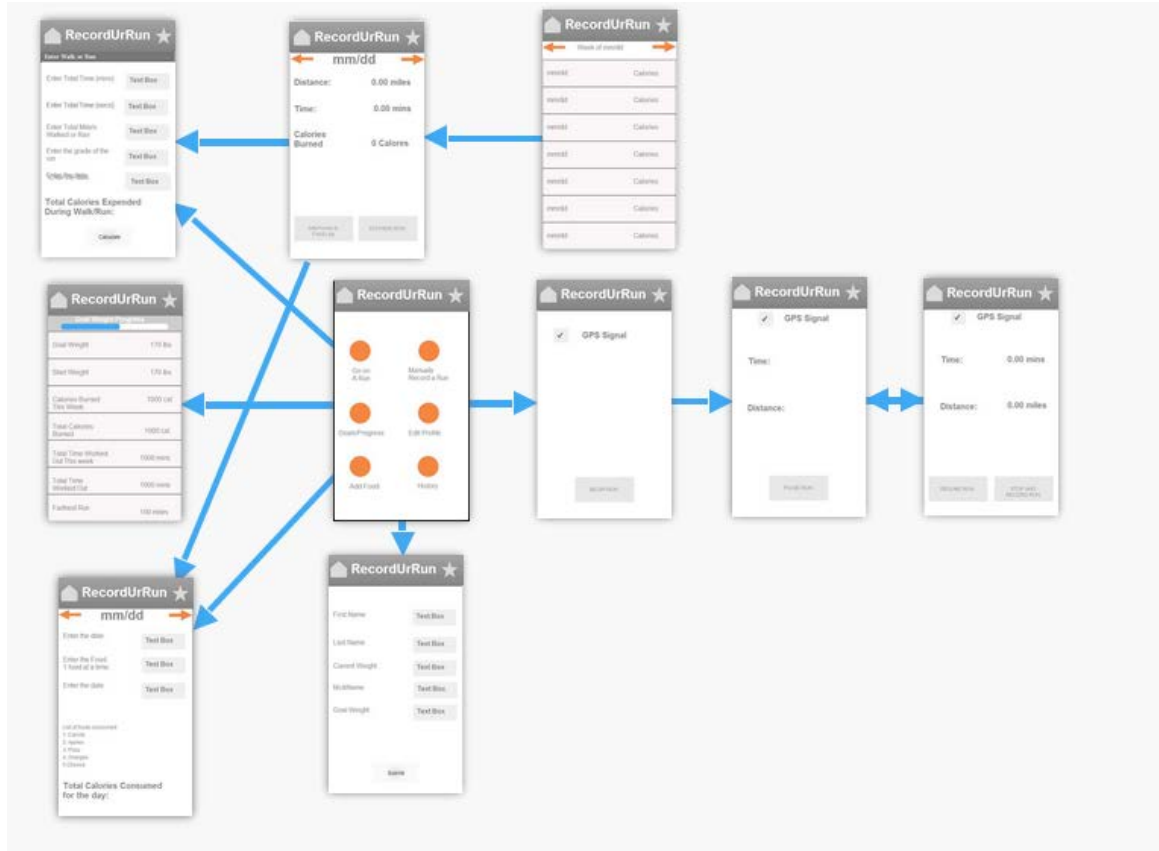
Time: 0.00 mins

Distance: 0.00 miles

RESUME RUN **STOP AND RECORD RUN**

RecordUrRun

- Go on A Run
- Manually Record a Run
- Goals/Progress
- Edit Profile
- Add Food
- History



Appendix E
Project Screen Shots

